Hit List

Clear | Generate Collection | Print | Fwd Refs | Bkwd Refs | Generate OACS

Search Results - Record(s) 1 through 1 of 1 returned.

☑ 1. Document ID: US 20010042190 A1

L39: Entry 1 of 1 File: PGPB Nov 15, 2001

PGPUB-DOCUMENT-NUMBER: 20010042190
PGPUB-FILING-TYPE: new
DOCUMENT-IDENTIFIER: US 20010042190 A1

TITLE: LOCAL AND GLOBAL REGISTER PARTITIONING IN A VLIW PROCESSOR

PUBLICATION-DATE: November 15, 2001

INVENTOR-INFORMATION:

NAME	CITY	STATE	COUNTRY	RULE-47
TREMBLAY, MARC	MENLO PARK	CA	US	
JOY, WILLIAM	ASPEN	CO	US	

APPL-NO: 09/ 204585 [PALM]
DATE FILED: December 3, 1998

CONTINUED PROSECUTION APPLICATION: CPA

INT-CL: [07] G06 F 15/00, G06 F 9/26, G06 F 12/00

US-CL-PUBLISHED: 712/208; 711/209
US-CL-CURRENT: 712/208; 711/209

REPRESENTATIVE-FIGURES: 2

ABSTRACT:

A Very Long Instruction Word (VLIW) processor having a plurality of functional units includes a multi-ported register file that is divided into a plurality of separate register file segments, each of the register file segments being associated to one of the plurality of functional units. The register file segments are partitioned into local registers and global registers. The global registers are read and written by all functional units. The local registers are read and written only by a functional unit associated with a particular register file segment. The local registers and global registers are addressed using register addresses in an address space that is separately defined for a register file segment/functional unit pair. The global registers are addressed within a selected global register range using the same register addresses for the plurality of register file segment/functional unit pairs. The local registers in a register file segment are addressed using register addresses in a local register range outside the global

register range that are assigned within a single register file segment/functional unit pair. Register addresses in the local register range are the same for the plurality of register file segment/functional unit pairs and address registers locally within a register file segment/functional unit pair.

Term	Documents
US-20010042190-A1	1
US-20010042190-A1.DID..PGPB.	1
(US-20010042190-A1.DID.).PGPB.	1

First Hit

End of Result Set

Generate Collection Print

L39: Entry 1 of 1 File: PGPB Nov 15, 2001

PGPUB-DOCUMENT-NUMBER: 20010042190
PGPUB-FILING-TYPE: new
DOCUMENT-IDENTIFIER: US 20010042190 A1

TITLE: LOCAL AND GLOBAL REGISTER PARTITIONING IN A VLIW PROCESSOR

PUBLICATION-DATE: November 15, 2001

INVENTOR-INFORMATION:

NAME	CITY	STATE	COUNTRY	RULE-47
TREMBLAY, MARC	MENLO PARK	CA	US	
JOY, WILLIAM	ASPEN	CO	US	

US-CL-CURRENT: 712/208; 711/209

CLAIMS:

What is claimed is:

1. A processor comprising: a plurality of functional units; and a register file that is divided into a plurality of register file segments, ones of the plurality of register file segments being coupled to and associated with ones of the plurality of functional units, the register file segments being partitioned into global registers and local registers, the global registers that are accessible by the plurality of functional units, the local registers being accessible by the functional unit associated with the register file segment containing the local registers.

2. A processor according to claim 1 wherein: the processor is a Very Long Instruction Word (VLIW) processor.

3. A processor according to claim 1 wherein: the local registers and global registers are addressed using register addresses in an address space that is defined for a register file segment/functional unit pair.

4. A processor according to claim 1 wherein: the register file is a multi-ported register file.

5. A processor according to claim 1 wherein: the local registers in a register file segment are addressed using register addresses in a local register range outside the global register range that are assigned within a single register file segment/functional unit pair.

6. A processor according to claim 1 wherein: register addresses in the local register range are the same for the plurality of register file segment/functional unit pairs and address registers locally within a register file segment/functional

unit pair.

7. A processor according to claim 1 wherein: the register file includes N physical registers and is duplicated into M register file segments, the register file segments having a reduced number of read and/or write ports in comparison to a nonduplicated register file, but each having the same number of physical registers.

8. A processor according to claim 7 wherein: the register file segments are partitioned into NG global and NL local register files where N.sub.G plus N.sub.L is equal to N, the register file operating equivalently to a register file having N.sub.G+(M*N.sub.L) total registers available for the M functional units, the number of address bits for addressing the N.sub.G+(M*N.sub.L) total registers being equal to the number of bits B that are used to address N=2.sup.B registers, the local registers for ones of the M register file segments are addressed using the same B-bit values.

9. A processor according to claim 6 wherein: partitioning of the register file is programmable so that the number N.sub.G of global registers and number N.sub.L of local registers is selectable and variable.

10. A processor according to claim 1 wherein the register file is a storage array structure having R read ports and W write ports comprising: a plurality of storage array storages; the storage array storages having a reduced number of read ports so that the total number of read ports for the plurality of storage array storages is R read ports; and the storage array storages having W write ports.

11. A processor according to claim 10 wherein: the storage array structure is a sixteen port structure with twelve read ports and five write ports; and the plurality of storage array storages includes four storage array storages each having three read ports and five write ports.

12. A processor according to claim 10 wherein: the storage array structure is a sixteen port structure with twelve read ports and four write ports; and the plurality of storage array storages includes four storage array storages each having three read ports and four write ports.

13. A processor according to claim 10 wherein: the writes are fully broadcast so that all of the storage array storages are held coherent.

14. A processor according to claim 10 wherein: storage array storages include storage cells having a plurality of word lines and a plurality of bit lines, the word lines being formed in one metal interconnect layer, the bits lines being formed in a second metal interconnect layer.

15. A processor comprising: a decoder for decoding a very long instruction word including a plurality of subinstructions, the subinstructions being allocated into positions of the instruction word; a register file coupled to the decoder and divided into a plurality of register file segments; and a plurality of functional units, ones of the plurality of functional units being coupled to an associated with respective ones of the register file segments, ones of the plurality of subinstructions being executable upon respective ones of the plurality of functional units, operating upon operands accessible to the register file segment associated with the functional unit of the plurality of functional units, the register file segments including a plurality of registers that are partitioned into global registers and local registers, the global registers being accessible by the plurality of functional units, the local registers in one of the register file segments being accessible by the functional unit associated with the register file segment.

16. A processor according to claim 15 wherein: the local registers and global registers are addressed using register addresses in an address space that is defined for a register file segment/functional unit pair.

17. A processor according to claim 15 wherein: the register file is a multi-ported register file.

18. A processor according to claim 15 wherein: the local registers in a register file segment are addressed using register addresses in a local register range outside the global register range that are assigned within a single register file segment/functional unit pair.

19. A processor according to claim 15 wherein: register addresses in the local register range are the same for the plurality of register file segment/functional unit pairs and address registers locally within a register file segment/functional unit pair.

20. A processor according to claim 15 wherein: the register file includes N physical registers and is duplicated into M register file segments, the register file segments having a reduced number of read and/or write ports in comparison to a nonduplicated register file, but each having the same number of physical registers.

21. A processor according to claim 20 wherein: the register file segments are partitioned into N.sub.G global and N.sub.L local register files where N.sub.G plus N.sub.L is equal to N, the register file operating equivalently to a register file having N.sub.G+(M*N.sub.L) total registers available for the M functional units, the number of address bits for addressing the N.sub.G+(M*N.sub.L) total registers being equal to the number of bits B that are used to address N=2.sup.B registers, the local registers for ones of the M register file segments are addressed using the same B-bit values.

22. A processor according to claim 20 wherein: partitioning of the register file is programmable so that the number N.sub.G of global registers and number N.sub.L of local registers is selectable and variable.

23. A method of operating a processor comprising: operating a plurality of functional units; and dividing a register file into a plurality of register file segments; coupling and associating ones of the plurality of register file segments with ones of the plurality of functional units; partitioning the register file segments into global registers and local registers; accessing the global registers by the plurality of functional units; accessing the local registers by the functional unit associated with the register file segment containing the local registers.

24. A method according to claim 23 further comprising: addressing the local registers and global registers using register addresses in an address space that is defined for a register file segment/functional unit pair.

25. A method according to claim 23 further comprising: addressing the local registers in a register file segment using register addresses in a local register range outside the global register range that are assigned within a single register file segment/functional unit pair.

26. A method according to claim 23 further comprising: addressing the local register range the same for the plurality of register file segment/functional unit pairs and address registers locally within a register file segment/functional unit pair.

27. A method according to claim 23 further comprising: including N physical registers in the register file; duplicated the physical registers into M register file segments, the register file segments having a reduced number of read and/or write ports in comparison to a nonduplicated register file, but each having the same number of physical registers.

28. A method according to claim 27 further comprising: partitioning the register file segments into N.sub.G global and N.sub.L local register files where N.sub.G plus N.sub.L is equal to N; operating the register file equivalently to a register file having N.sub.G+(M*N.sub.L) total registers available for the M functional units, the number of address bits for addressing the N.sub.G+(M*N.sub.L) total registers being equal to the number of bits B that are used to address N=2.sup.B registers; and addressing the local registers for ones of the M register file segments using the same B-bit values.

29. A method according to claim 27 further comprising: programmably partitioning the register file so that the number N.sub.G of global registers and number N.sub.L of local registers is selectable and variable.

First Hit Fwd Refs

Generate Collection Print

L3: Entry 10 of 11 File: USPT Jul 9, 2002

US-PAT-NO: 6418527
DOCUMENT-IDENTIFIER: US 6418527 B1
** See image for Certificate of Correction **

TITLE: Data processor instruction system for grouping instructions with or without a common prefix and data processing system that uses two or more instruction grouping methods

DATE-ISSUED: July 9, 2002

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Rozenshein; Zvika	Kfar Saba			IL
Tokar; Jacob	Ashdod			IL
Dayan; Uri	Herzelia			IL
Gergen; Joe Paul	Austin	TX		

ASSIGNEE-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY	TYPE CODE
Motorola, Inc.	Schaumburg	IL			02

APPL-NO: 09/ 170690 [PALM]
DATE FILED: October 13, 1998

INT-CL: [07] G06 F 15/00

US-CL-ISSUED: 712/208; 712/204, 712/205, 712/206, 712/209, 712/212, 712/213, 712/214, 712/215
US-CL-CURRENT: 712/208; 712/204, 712/205, 712/206, 712/209, 712/212, 712/213, 712/214, 712/215

FIELD-OF-SEARCH: 712/204, 712/208, 712/212, 712/215, 712/205, 712/206, 712/209, 712/213, 712/214

PRIOR-ART-DISCLOSED:

U.S. PATENT DOCUMENTS

Search Selected Search ALL Clear

	PAT-NO	ISSUE-DATE	PATENTEE-NAME	US-CL
☐	5560028	September 1996	Sachs et al.	395/800
☐	5630083	May 1997	Carbine et al.	712/212

☐	5655097	August 1997	Witt et al.	712/204
☐	5689672	November 1997	Witt et al.	712/213
☐	5732234	March 1998	Vassiliadis et al.	712/200
☐	5794003	August 1998	Sachs	712/215
☐	5796973	August 1998	Witt et al.	712/208
☐	5822778	October 1998	Dutton et al.	711/208
☐	6044450	March 2000	Tsushima et al.	712/24
☐	6275927	August 2001	Roberts	712/213

OTHER PUBLICATIONS

Nov., 1998 IEEE Spectrum, vol. 35, No. 11, "Engineering the EV future", Article: High-flying DSP architectures, pp. 53-56. (Resubmit).

ART-UNIT: 2783

PRIMARY-EXAMINER: Maung; Zarni

ASSISTANT-EXAMINER: El-Hady; Nabil

ATTY-AGENT-FIRM: King; Robert L.

ABSTRACT:

A system for instructing a data processor, the system including an instruction root having an operation selection field for selecting an operation to be performed by said data processor and an instruction prefix. The instruction prefix has a field selected from the group of a conditional execution field for selecting a condition under which a data processor will perform said selected operation, an operand length modification field for modifying the selected operation so as to be performed on an operand having a different length, an instruction group field for selecting a length of an instruction group that includes the instruction root, and a prefix length selection field for selecting a length of said instruction prefix. A data processor system responsive to this instruction system is also disclosed. An instruction system for statically grouping instructions without using an instruction prefix is also disclosed.

12 Claims, 19 Drawing figures

L3: Entry 10 of 11 File: USPT Jul 9, 2002

DOCUMENT-IDENTIFIER: US 6418527 B1
** See image for Certificate of Correction **
TITLE: Data processor instruction system for grouping instructions with or without a common prefix and data processing system that uses two or more instruction grouping methods

Detailed Description Text (11):
Referring to FIG. 2, a particular embodiment of registers within the core 12 of the system 10 is disclosed. As illustrated, the address register file 26 includes registers R0-R7, stack pointer (SP), N0-N3, M0-M2, MCTL, SA0-SA3, LC0-LC3. The program sequencer 24 includes the program counter, status register, and operating mode and status registers. The data register file 28 includes registers D0-D7 and the data register file 29 includes registers D8-D15. In an alternative embodiment, only a single register file may be used to save cost, such as with the one or two MAC configurations. In other high performance applications, more than two register files may also be used.

WEST Search History

Hide Items | Restore | Clear | Cancel

DATE: Sunday, April 18, 2004

Hide?	Set Name	Query	Hit Count
	DB=PGPB,USPT; PLUR=NO; OP=ADJ		
☐	L3	l1 same L2	11
☐	L2	"two register files"	150
☐	L1	"single register file"	139

END OF SEARCH HISTORY

WEST Search History

Hide Items | Restore | Clear | Cancel

DATE: Sunday, April 18, 2004

Hide?	Set Name	Query	Hit Count
	DB=USPT; PLUR=NO; OP=ADJ		
☐	L31	synchroniz$5 and L29	10
☐	L30	synchriniz$5 and L29	0
☐	L29	vliw and L28	29
☐	L28	5574939.uref.	53
	DB=EPAB,JPAB,DWPI,TDBD; PLUR=NO; OP=ADJ		
☐	L27	5574939.uref.	0
☐	L26	clustered vliw	0
☐	L25	multiple vliw cores	0
☐	L24	multiple vliw cores	0
☐	L23	vliw and L22	10
☐	L22	multiprocessor or multi-processor or (multiple processors)	12023
	DB=PGPB,USPT; PLUR=NO; OP=ADJ		
☐	L21	L6 with vliw	44
☐	L20	clustered vliw	2
☐	L19	variable issue width	10
☐	L18	synchroniz$5 same L17	4
☐	L17	vliw processors	254
☐	L16	multiple vliw cores	1
☐	L15	multiple vliw processors	0
☐	L14	multiple vliw pipelines	0
☐	L13	vliw same L12	21
☐	L12	multiple pipelines	384
☐	L11	vliw and L10	5
☐	L10	5459798.uref.	21
☐	L9	vliw same L8	4
☐	L8	L6 with chip	892
☐	L7	L6 adj chip	47
☐	L6	multiprocessor or multi-processor or (multiple processors)	22608
☐	L5	vliw and L4	6
☐	L4	pipeline paths	24
☐	L3	l1 same L2	11

☐	L2	"two register files"	150
☐	L1	"single register file"	139

END OF SEARCH HISTORY